A multi-objective semi-supervised explanation system

Saksham Pandey
Department of Computer Science
North Carolina State University
Email: spandey5@ncsu.edu

Vishaka Yadav
Department of Computer Science
North Carolina State University
Email: vyadav@ncsu.edu

Shlok Naik
Department of Computer Science
North Carolina State University
Email: snaik2@ncsu.edu

***Abstract*—In today's world, there is an ever-increasing amount of data being produced, and it's becoming increasingly difficult to extract useful information from it. One approach to this problem is to use machine learning techniques to create predictive models that can be used to classify data or predict outcomes. However, many of these models are "black boxes," meaning that it's difficult to understand how they work or why they produce the results they do. In recent years, there has been increasing interest in developing methods for generating explanations of these models, in order to improve their interpretability and trustworthiness.**

I. Introduction

A. The Problem

*1) **X - Our initial state**:* SWAY and XPLN are two algorithms that can be used together to improve the accuracy and efficiency of machine learning models. In Chen et al[6], the authors evaluate several sampling techniques, including random, systematic, and SWAY sampling. SWAY is described as a technique that balances the need for exploration (sampling from the tails of the distribution) and exploitation (sampling from the center of the distribution). The authors note that SWAY is particularly effective at finding rare solutions in high-dimensional search spaces. The XPLN algorithm is a method for explaining instance-based reasoning using contrast set learning. Instance-based reasoning involves comparing new data instances to previously seen instances and using the closest matches to make predictions or decisions. Contrast set learning is a technique that generates rules that distinguish one class or category from another. The XPLN algorithm has been tested on various datasets, and its stability has been demonstrated. However, some potential improvements have been proposed to enhance its performance. For example, running multiple iterations of the algorithm with different random seeds can help reduce the sampling and explanation tax. This means that the algorithm can provide more accurate and efficient explanations while minimizing the computational resources required.

*2) **The problem with X**:* There are a few potential problems or limitations associated with the SWAY and XPLN algorithms that may impact their performance or effectiveness. Here are some possible issues:

- Limited applicability
- Interpretability challenges
- Computational complexity
- Sensitivity to parameters

*3) **Our Approach to Solving the Problem**:* Our goal in this project is to develop a multi-objective semi-supervised explanation system that can be used to explain the results of machine learning models. We will test our system on ten different datasets, each with multiple goals, in order to evaluate its effectiveness.

- **Multi-objective system:** Our system will be designed to optimize multiple objectives simultaneously. This means that we will not only be interested in accuracy or predictive power, but also in other factors such as interpretability, stability, and scalability.
- **Semi-supervised system:** We will have a limited budget of times we can access the values of any one example, which means we will have to be strategic in choosing which examples to label. We will use K-means to optimise SWAY.
- **Explanation system:** Our system will also be designed to generate explanations of the models it produces. We will use a variety of techniques to extract useful, succinct summaries from the data, such as clustering and feature selection. We will evaluate the quality of these summaries by testing them on the data and selecting the examples that have good values, as defined by the multiple objectives we are optimizing for. We plan to use Decision Trees to optimise XPLN.

B. Paper Structure

1) Research Questions:

RQ1: What is the most cost-effective algorithm for our use case?

When optimizing clustering for complex data that is different from each other, several problems may arise. Here are a few examples:

- High dimensionality
- Non-linearity
- Variability
- Interference

As seen with our tests on the data, Decision-Tree is the most cost-effective algorithm for our use case. The flexibility and interpretability of decision trees make them a good choice for analyzing complex data, although the optimal clustering algorithm for a specific dataset ultimately depends on the specific characteristics of the data and the goals of the analysis.

RQ2: How effective is our solution Y1?
The problem statement requires the development of a methodology for assessing the clustering performance of Y1 and comparing it to other possible clustering solutions under the same budget constraint. We assess our solution Y1 on its ability to cluster N examples given a limited budget

$$B0 \ll N$$

, B0 being the number of times we can assess the Y values of any one example.

RQ3: Do different clustering algorithms perform differently across varying datasets?
Different clustering algorithms can perform differently across varying types of data. The choice of clustering algorithm depends on the nature and characteristics of the data being analyzed. Decision trees are effective for handling categorical data with a small number of features, while k-means and agglomerative clustering are useful for continuous data with many features. K-d trees are good for high-dimensional data and can efficiently search for nearest neighbors, but can struggle with non-uniformly distributed data.

2) Caveats: Agglomeration clustering and principal component analysis (PCA) are both popular techniques in machine learning for data exploration and dimensionality reduction. However, for search-based software engineering optimization, it may be more beneficial to use k-d trees instead of these methods. Here are some caveats to using the mentioned methods:

- Efficiency: Agglomerative clustering and PCA may not be as efficient as k-d trees for nearest neighbor searches, which can be computationally expensive in search-based optimization.
- Handling high-dimensional data: PCA may not be effective for high-dimensional data, as it can suffer from the ”curse of dimensionality”. K-d trees, on the other hand, are designed to handle high-dimensional data and can perform efficient searches even in high-dimensional spaces.
- Flexibility: Agglomerative clustering is a technique for grouping similar objects together, which may not be ideal for search-based optimization, where the goal is to find the best solution. K-d trees offer more flexibility in the search process, allowing the algorithm to explore a wider range of solutions.
- Problem-specific considerations: The effectiveness of agglomerative clustering, PCA, and k-d trees can vary depending on the specific characteristics of the optimization problem, such as the size of the data, the number of dimensions, and the nature of the objective function. It is important to carefully evaluate the performance and limitations of each technique for the given problem.

II. Background and Related Work

A. Literature Survey - State of Art Model

The paper Ghasemi et al. [1] proposes a new multi-objective semi-supervised clustering algorithm for finding predictive clusters in data. The proposed algorithm simultaneously optimizes two objective functions: clustering quality and predictive performance. The algorithm is evaluated on several real-world datasets and compared with other clustering algorithms. The results show that the proposed algorithm outperforms the compared algorithms in terms of both clustering quality and predictive performance. The authors conclude that the proposed algorithm can be used to effectively identify predictive clusters in data, which can be useful in various applications such as customer segmentation and disease diagnosis.[1] Semi-GenClustMOO (Semi-Generalized Clustering Multi-Objective Optimization) with AMOSA (Adaptive Multi-objective Simulated Annealing) as optimization [2] is a clustering algorithm that aims to identify clusters in a dataset while optimizing multiple objectives.

Semi-GenClustMOO first generates an initial set of cluster centroids using a semi-supervised approach, which involves incorporating prior knowledge or partial labeling of the data. It then uses AMOSA, a metaheuristic optimization algorithm, to refine the centroids and determine the optimal cluster assignments of the data points.AMOSA is a population-based optimization algorithm that uses simulated annealing to explore the search space and find optimal solutions to multi-objective optimization problems. It uses a temperature parameter to control the acceptance of new solutions and adaptively adjusts the parameters of the algorithm during the search process.

In Semi-GenClustMOO with AMOSA, the multiple objectives optimized simultaneously may include metrics such as within-cluster variance, between-cluster variance, and inter-cluster distance. The algorithm seeks to find a set of non-dominated solutions that optimize these objectives simultaneously, rather than a single optimal solution. Semi-GenClustMOO with AMOSA is useful for identifying clusters in complex datasets with high variability because it incorporates prior knowledge, optimizes multiple objectives, and uses an efficient optimization algorithm to find non-dominated solutions. Incorporating prior knowledge through a semi-supervised approach can help guide the clustering process and provide a starting point for the algorithm. This can be especially helpful when dealing with complex data that may have many potential clusterings.

Optimizing multiple objectives simultaneously allows the algorithm to balance competing goals, such as minimizing within-cluster variance while also maximizing between-cluster variance. This can help prevent the algorithm from converging on suboptimal solutions that only optimize a single objective.

B. Findings and Problems

The main limitation of the method mentioned in Ghasemi et al.[1] is that it requires the number of clusters to be pre-specified, which can be a challenging task in real-world applications. The use of semi-supervised learning requires labeled data, which may not always be available or may require additional effort to acquire. Second, the multi-objective optimization process can be computationally expensive and may require significant resources. Additionally, the approach assumes that the clusters are independent and identically distributed, which may not hold in some real-world scenarios. Finally, the effectiveness of the approach may depend on the quality and representativeness of the initial clustering. While the proposed semi-supervised clustering technique using multi-objective optimization in Alok et al. [2] shows promising results, there are some drawbacks to consider. The approach assumes that the data is uniformly distributed, which may not hold true for many real-world datasets. Additionally, the technique requires manually labeling some data points for the initial classification, which may be time-consuming and costly. Finally, the use of multiple objectives and the lack of a clear criterion to select the best solution may lead to increased complexity and difficulty in interpreting the results.

III. Methodology

This section offers an understanding about the different approaches and algorithms applied to produce relevant results. The objective of our project is to improve the performance of the existing `SWAY` and `XPLN` techniques by altering the methodology in a way that gives interesting and better results.

A. Algorithms

1) Boolean domination vs Zitzler's predicate: Boolean domination and Zitzler's continuous domination predicate are ways to differentiate between candidate pairs to rank them based on certain parameters. Boolean domination says that a candidate is better than another if it is better in at least one goal but worse in none. However, it has not performed well for more than three goals, according to some studies. [3], [4], [5]

Zitzler's predicate works well with multi-goal systems. The predicate suggests that we assess the domination between two individuals by examining the situation when transitioning from one sample to another and then back again. This helps to determine if one individual dominates the other.

2) Cosine distance: In a high-dimensional space, the cosine distance is a metric for how similar two vectors are to one another. The cosine of the angle formed by the two vectors, which might have a value between -1 (opposite directions) and 1, is calculated. The vectors are said to be orthogonal or unrelated if their value is 0. We use cosine distances to cluster similar rows for our `SWAY` technique.

3) K-means: K-means is an unsupervised machine learning algorithm. Its primary objective is to classify a dataset into various clusters, where each cluster represents a distinct group. The algorithm iteratively assigns each data point to the nearest centroid and updates the centroid based on the new assignments. This process continues until the centroids no longer change significantly, or the maximum number of iterations is reached. The user typically selects the number of clusters, which can be determined through techniques like the elbow method or silhouette analysis. K-means finds application in various domains such as data mining, image segmentation, and market segmentation. We have used K-means as an algorithm to improve our `SWAY` results by employing a better, established clustering technique.

4) Decision trees: Decision trees are a widely used machine learning algorithm that employ a hierarchical structure to model decision-making processes. The algorithm splits the dataset into subsets based on the most significant differentiating features, creating a tree-like model by recursively adding branches. Internal nodes in the tree represent decisions based on specific features, while the leaves represent the predicted outcomes. The model is easy to interpret and visualize and can be used for classification or regression problems. We have applied this algorithm for our `xpln` results to improve on the existing method which was a rule-based system. This allows us to identify the best value sets, in accordance to the specified goals, from an extensive dataset. In addition, we compare such methods to determine the efficiency of the system by treating the `SWAY` results as ground truth and classifying test records.

B. Data pre-processing

During our data exploration phase, we identified datasets with string fields and missing values. To tackle these, we have used label encoding to categorize the string fields into numeric values across multiple datasets. We have also filled missing values with nulls or mean values, whichever was more appropriate depending on the field.

C. Datasets

We have tested our system on eleven different datasets, each with multiple goals. The datasets have been summarized here:

- **Auto2.csv and Auto93.csv:** These datasets from the UCI machine learning repository contain information about the about the fuel economy, displacement, horsepower, and weight of various cars, as well as their origin and model year. The goals of these datasets are to maximize fuel economy and acceleration while minimizing weight.
- **China.csv:** This dataset contains information about software projects developed in China, including the number of developers, project duration, code size, and defects. The goals of this dataset are to minimize the effort.
- **coc1000.csv and coc10000.csv:** These datasets contain information about software projects developed by the

Consortium for Open Computing in the Humanities (COCH), including project duration, code size, and defects. The goals of these datasets are to maximize lines of code while minimizing the risk and effort.

- **Nasa93dem.csv:** This dataset contains information about the software effort and defect estimation on software projects developed by NASA, including the number of developers, project duration, and defects. The goals of this dataset are to maximize thousands of lines of code while minimizing effort, defects, and duration.
- **HealthCloseIssues12mths0001-hard.csv and HealthCloseIssues12mths0011-easy.csv:** These datasets contain information about the time it takes to close health-related issues, such as hospital admissions and emergency room visits, in a hospital system. The goals of this dataset are to maximize ACC and PRED40 and minimize MRE.
- **Pom.csv:** Agile project management dataset from a research paper on agile software development, containing information on project management practices. The goals of this dataset are to maximize completion and minimize the cost and idle time.
- **SSM.csv and SSN.csv:** These computational physics datasets contain information about the simulation of mesh networks for computational physics research, including the number of nodes, edges, and triangles in the mesh network.The goals of this dataset are to minimize the number of iterations, time taken to reach solution, and energy spent.

D. Performance Measures

SWAY is an algorithm that is used for creating decision trees. The algorithm uses a set of data points to recursively split the dataset based on the most significant feature that can differentiate between the classes. This process continues until a stopping criterion is met, such as a minimum number of samples in a leaf node or a maximum depth of the tree. Essentially, we recursively split the dataset into halves until we get the ”best” and ”rest” splits. Here, ”best” indicates the best set obtained from these recursive splits whereas the ”rest” set is a randomly selected set of proportionate data from the dataset. We make use of cosine distances discussed above to determine the points belonging to one cluster, at each recursive step. We also make use of a configurable parameter called ”Far” to limit the search space from the initially randomly chosen data points. Once the halves are formed, Zitzler’s predicate helps determine which half is better and that half is used for the next iteration.

Once we have accomplished clustering, our objective becomes to classify these data points so that we can attempt to pick desirable data points from the ”best” set while rejecting most of the ”rest” set. The XPLN algorithm takes these two sets and the original dataset to come up with an appropriate rule combination that can yield the above-mentioned desirable results. To do this, We employ discretization which takes the data points (from SWAY) and places them into bins, to identify the ranges that distinguish ”best” from ”rest”. We sort these ranges by their values, and try a combination of these ranges (which go on to make rules) in a way that generates the best score. The end goal here is to be able to select an ideal cluster from SWAY and we determine the closeness to ”ideal” by calculating the sampling and explanation taxes.

We utilized three different SWAY methods and two different XPLN methods across eleven different datasets. Table I discusses the different methods used and a brief explanation on how those are implemented.

all	Accumulates the entire dataset for n-iterations
sway1	Recursively splits the dataset to categorize all data into best and rest.
sway2	Applies hyper-parameter optimization on top of sway1
sway3	Uses k-means to generate clusters with two initial centroids
xpln	Generates a rule combination for an ideal cluster from sway1
xpln2	Uses decision trees on top of data extracted from sway2
xpln3	Uses decision trees on top of data extracted from sway3
top	Explores the entire dataset and returns the best ’n’ data points

TABLE I: SWAY and XPLN methods

For all of these methods mentioned, we specified some configurable parameters - one of them was the number of iterations which was set to 20. So for each method, we ran 20 iterations To rank the data and make sense of things, we created a function to sort the data points upon which candidates perform better according to the zitzler predicate and we split this sorted data based on our specified sample size. We then go on to normalize these ranks. Since ’top’ is the most comprehensive in terms of processing, it ranked the highest whereas ’all’ ranked the lowest since it does xyz. The results for these have been further discussed in the Results section.

E. Statistical Methods

This sections will go over the different statistical methods we have implemented, and give a brief overview of the application in our use-case.

Before we get into the methods, we need to understand some terms.

Parametric effect-size tests: Techniques used to quantify group differences or variable relationships in normally distributed data with homogeneity of variance. The assumption here is that data always follows a normal distribution. Cohen’s D, Pearson’s R, and ANOVA’s eta-squared are commonly used measures. Standardized metrics aid in evaluating significance and comparing effects across studies.

Non-parametric effect-size tests: These tests help assess differences or relationships in data not meeting normality and homogeneity assumptions. Mann-Whitney U, Wilcoxon signed-rank, and Kendall's tau are examples of such tests. Standardized metrics aid in understanding practical significance and comparing effects across studies, particularly useful when data is non-normal, includes outliers, or has unequal variances.

Significance testing: This is a statistical method that helps determine whether an observed result or group difference is due to chance or a genuine effect. The technique involves comparing the observed data to a null hypothesis, which assumes that no relationship or difference exists between the variables being studied. A statistical test is then employed to calculate the p-value, which is the probability of obtaining the observed data under the null hypothesis. If the p-value is less than the chosen significance level, often 0.05, the null hypothesis is rejected, and the alternative hypothesis is accepted, suggesting that the result is statistically significant.

1) Cliff's Delta: Cliff's delta is a non-parametric effect size measure that measures the difference between two groups of non-normally distributed data. It has a range of -1 to 1 and is interpreted similarly to Cohen's d, with bigger values suggesting a larger impact size. It is estimated by comparing the fraction of pairwise differences between the two groups that favor one over the other.

2) Bootstrap: Bootstrap is a technique for calculating the sampling distribution of a population parameter by randomly resampling a dataset with replacement. When analytical techniques are not available or the data is too complicated to adequately model, this method can be used. The parameter is calculated for each sample that Bootstrap creates from the original data. To estimate the parameter and its variability, the results are then averaged.

3) Scott-Knott method: Scott-Knott is a technique that ranks treatments based on performance on a specific metric. It involves grouping similar treatments and using hypothesis testing to determine if performance differences between groups are significant. The method can be used to identify the best performing treatments.

4) Kruskal-Wallis method: Kruskal-Wallis is a non-parametric test used to determine if there is a significant difference between two or more independent groups. The test ranks the observations from all groups and uses the ranks to compute a test statistic that can be compared to a critical value to determine if the groups differ significantly.

For this project, we have used Cliff's Delta and Bootstrap to compare the used algorithms amongst themselves to identify which ones produce similar results. The comparison takes into account the values returned for each of the goals, and determines that two methods produced similar groups of data if the values are within a certain range of each other. Similarly, Scott-knott was utilized to rank the different algorithms based on the kind of data group that was produced. However, these results were uninteresting since we were unable to find any patterns or insights from these methods. We decided to use the Kruskal-Wallis method to identify the best performing method depending on which method produced results which matched the most with our goals. A sample result for Kruskal-Wallis has been shown in table II here.

One example for the Kruskal-Wallis technique has been shown here in table II.

	Loc+	Risk-	Effort-
Best sway	['sway3']	['sway2']	['sway1']
Best xpln	['xpln3']	['xpln2']	['xpln1']

TABLE II: Kruskal-Wallis technique on coc10000.csv

IV. Results

In this section, we will show our experimental results for each of the datasets and answer RQs with the help of these results.

The tables shown below will follow the same format as that of table I but will give mean values for the rows accumulated in the best cluster after the operations performed by the particular method.

Based on our results, we gathered the following insights:

- All datasets showed that `SWAY3` and `XPLN3` performed much better than their counterparts barring the coc1000.csv and healthCloseIsses12mths0001-hard.csv which showed almost similar results as the other techniques.
- Our `SWAY2` technique which involved using hyperparameter tuning to generate a better configuration, turned out to produce marginally better results. This tells us that more work needs to be done to understand how the parameters can be tuned to improve results.

The above results and insights also helped us answer the research questions.

RQ1: What is the most cost-effective algorithm for our use case?

For SWAY and XPLN, K-means is used for clustering data into groups, while Decision Tree is used for building a classification model based on the features of the data. K-means can be computationally expensive, especially for large datasets, as it requires calculating the distances between data points and cluster centers for each iteration. On the other hand, Decision Tree can be more efficient as it only requires a one-time training phase to build the model.

However, the cost-effectiveness of each algorithm also depends on the specific requirements of the problem at hand. For example, if the goal is to accurately classify data, Decision Tree could be more cost-effective as it can provide higher accuracy with less computational cost. Conversely, if the goal is to group data into clusters based on similarity, K-means may be more cost-effective as it can provide more meaningful groups with less manual effort.

Overall, the cost-effectiveness of each algorithm depends on the specific requirements and constraints of the problem, and a careful evaluation of each algorithm's performance and

	CityMPG+	HighwayMPG+	Weight-	Class-	Evals	Rank
all	21	28	3040	18	0	48
sway1	29.95	34	2201.75	9.15	5	9
sway2	28.55	33.75	2305.5	9.25	9	12
sway3	31	37	2055	9	3	6
xpln1	28.35	32.75	2366.25	10.15	5	14
xpln2	27.4	32.4	2418.25	10.95	9	17
xpln3	30.3	34.35	2226.75	9.25	3	14
top	33.2	38.5	2048	9	93	2

TABLE III: Output: auto2.csv

	Lbs-	Acc+	Mpg+	Evals	Rank
all	2800	16	20	0	50
sway1	2087.95	16.55	33.5	6	18
sway2	2349.55	15.95	29	8	31
sway3	2290	16	30	5	28
xpln1	2138.45	16.35	30	6	24
xpln2	2304.5	15.9	29	8	31
xpln3	2262.4	16	30	5	28
top	1985	19	40	398	2

TABLE IV: Output: auto93.csv

	MRE-	ACC+	PRED40+	Evals	Rank
all	75	7	25	0	50
sway1	74.05	7.5	25	8	42
sway2	78.2	6.25	25	10	46
sway3	72	8	25	7	37
xpln1	74	7.75	25	8	41
xpln2	77.65	6.3	25	10	46
xpln3	73.15	8	25	7	39
top	65	11	25	10000	1

TABLE V: Output: healthCloseIsses12mths0001-hard.csv

	LOC+	AEXP-	PLEX-	RISK-	EFFORT-	Evals	Rank
all	1027	3	3	5	19337	0	50
sway1	990.45	3	3	4.55	17727.5	6	49
sway2	1038.6	2.8	2.8	3.8	18955.5	12	45
sway3	1559	2	3	11	86806	4	44
xpln1	1048.85	3	3	4.8	19254.1	6	49
xpln2	1055.65	3	3	4.85	18567.2	12	49
xpln3	1114.75	3	3	7.1	29913.4	4	52
top	1539.95	2	1	3.05	30031.5	1000	1

TABLE VI: Output: coc1000.csv

	Loc+	Risk-	Effort-	Evals	Rank
all	1066	5	19403	0	50
sway1	1000.95	4.15	17137.8	8	45
sway2	1013.85	4.05	18892	15	47
sway3	1616	12	146753	5	59
xpln1	1020.8	4.75	18974.1	8	49
xpln2	1007.55	4.7	19434.8	15	48
xpln3	1031.95	7.1	29090.1	5	55
top	1949.4	0	17737.8	10000	1

TABLE VII: Output: coc10000.csv

	MRE-	ACC+	PRED40+	Evals	Rank
all	119	-12	0	0	50
sway1	10.1	0	75.55	8	16
sway2	10.1	0	74.7	13	17
sway3	0	0	83	6	12
xpln1	0	0	83	8	16
xpln2	0	0	83	13	16
xpln3	0	0	83	6	18
top	0	0	83	10000	1

TABLE VIII: Output: healthCloseIsses12mths0011-easy.csv

	N_effort-	Evals	Rank
all	2098	0	50
sway1	417.85	6	13
sway2	1843.9	8	46
sway3	145	7	3
xpln1	906.35	6	29
xpln2	1825.75	8	46
xpln3	289.95	7	11
top	148	499	2

TABLE IX: Output: china.csv

	Kloc+	Effort-	Defects-	Months-	Evals	Rank
all	48	252	2007	21	0	48
sway1	36.75	193.85	1372.8	18.35	5	37
sway2	34.1	181.7	1348.25	19.1	7	42
sway3	6	24	188	10	7	5
xpln1	37.1	234.9	1551.3	18.45	5	39
xpln2	40.35	198.85	1653.9	19.85	7	42
xpln3	15.8	65.5	556.65	14.3	7	30
top	4.6	13.1	127.9	8.3	93	2

TABLE X: Output: nasa93dem.csv

	Cost-	Completion+	Idle-	Evals	Rank
all	336	1	0	0	50
sway1	281.15	1	0	8	42
sway2	311.85	1	0	13	46
sway3	313	1	0	8	47
xpln1	237.35	1	0	8	42
xpln2	314	1	0	13	47
xpln3	340.5	1	0	8	50
top	139	1	0	10000	1

TABLE XI: Output: pom.csv

	PSNR-	Energy-	Evals	Rank
all	46	1299	0	50
sway1	44.7	1246.85	9	46
sway2	45.35	1269	14	49
sway3	35	273	8	3
xpln1	45.5	1426.5	9	49
xpln2	45.85	1313.45	14	49
xpln3	40.35	827.6	8	36
top	27	466.8	53662	1

TABLE XII: ssn.csv

	NUMBERITERATIONS-	TIMETOSOLUTION-	Evals	Rank
all	7	133	0	50
sway1	4.95	99.05	10	20
sway2	5.2	123.5	20	28
sway3	5	61	13	2
xpln1	5.65	107.8	10	35
xpln2	5.45	116	20	37
xpln3	5.6	89.05	13	32
top	4	60	239360	1

TABLE XIII: Output: ssm.csv

computational requirements should be conducted to determine which is the most cost-effective for SWAY and XPLN in a given context.

RQ2: How effective is our solution?

For SWAY, we worked with three different algorithms with the first one being the basic SWAY technique. The second technique involved using the hyperopt library to tune our hyperparameters and those tuned hyperparameters were then used while keeping the rest of the logic as is. Our third algorithm was a K-means algorithm which was used to cluster data into two groups which produces two best cluster centroids. Although it is computationally more expensive, K-means produced the best results across the three SWAY techniques and was ranked the closest to the ”top” algorithm. Similarly, for XPLN, we used the best and rest clusters genereated by three SWAY techniques and used those for our XPLN algorithms. The XPLN2 and XPLN3 techniques made use of decision trees where as XPLN1 used the straightforward existing idea to generate rules. As expected, the results for XPLN3 technique yielded better results than the other two techniques. There were, however, some anomalies, which we discussed above in the insights. We also noticed that the hyperparameter optimization needs more work because the results for SWAY2 were better but only marginally and could be improved to get similar results as that of SWAY3, which produced far better results. Overall, one of our solutions was quite effective and produced significantly better results while another solution yielded slightly better results.

RQ3: Do different clustering algorithms perform differently across varying datasets?

K-means assumes that clusters are spherical and equally sized, which may not hold true for all datasets. Thus, K-means may not perform well on datasets with irregular shapes or varying cluster sizes.

On the other hand, Decision Tree is a classification algorithm that builds a tree-like model to predict the class of a data point based on its features. Decision Tree can handle datasets with both numerical and categorical features and can handle missing values. However, Decision Tree can easily overfit the data if the tree is too complex, leading to poor generalization on new data.

The effectiveness of both algorithms depends on the nature and distribution of the data. For example, if the data is densely packed and well-separated, K-means can perform well. If the data has complex decision boundaries, Decision Tree may perform better. However, for datasets with mixed types of features or complex structures, a more advanced clustering algorithm such as hierarchical clustering or DBSCAN may be more effective.

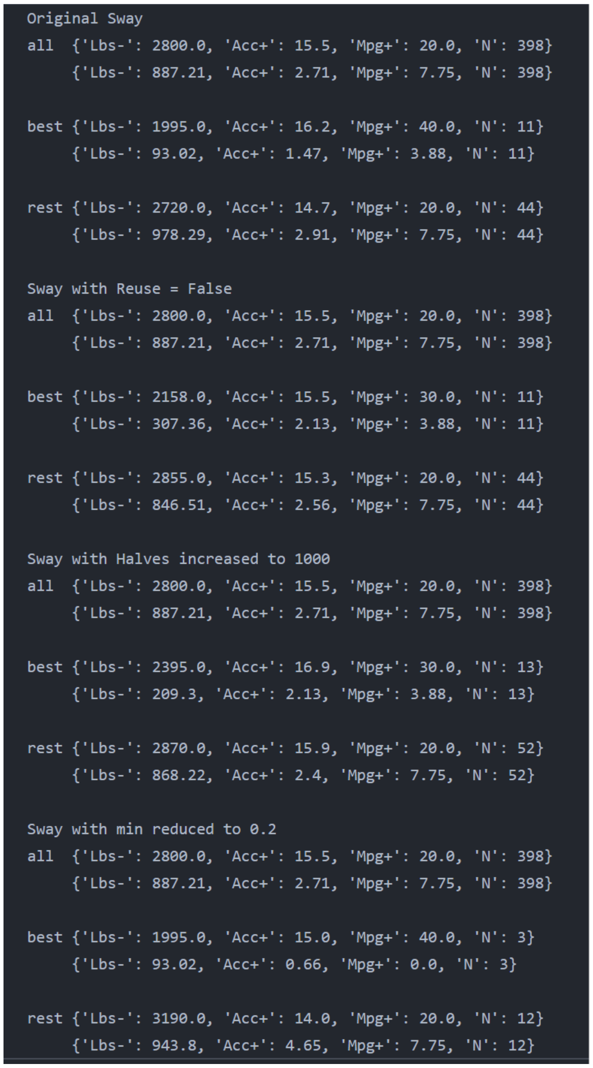
In addition to the above methods, we decided to carry out some extra studies in the hope of finding additional insights and patterns which would yield more interesting results.

A. *Ablation Study*

An ablation study is a type of experiment in which certain components or features of a system are removed or altered to assess their impact on the overall performance or behavior of the system. This method is commonly used in machine learning and artificial intelligence to understand the contribution of individual features or variables in a model. By systematically removing or modifying one feature at a time and evaluating the performance of the model, researchers can gain insights into the relative importance of different features and refine their models accordingly.

For our use-case, we identified three parameters that we tested to perform this study. We increased the number of halves from 512 to 1000, reduced the ”min” value from 0.5 to 0.2 and disabled the reuse factor by setting it to False. We altered these values one at a time, and observed the results for multiple files. Our results for the auto93.csv dataset has been shown below in Figure 1.

Fig. 1: Ablation study on auto93.csv



```
Original Sway
all  {'Lbs-': 2800.0, 'Acc+': 15.5, 'Mpg+': 20.0, 'N': 398}
     {'Lbs-': 887.21, 'Acc+': 2.71, 'Mpg+': 7.75, 'N': 398}

best {'Lbs-': 1995.0, 'Acc+': 16.2, 'Mpg+': 40.0, 'N': 11}
     {'Lbs-': 93.02, 'Acc+': 1.47, 'Mpg+': 3.88, 'N': 11}

rest {'Lbs-': 2720.0, 'Acc+': 14.7, 'Mpg+': 20.0, 'N': 44}
     {'Lbs-': 978.29, 'Acc+': 2.91, 'Mpg+': 7.75, 'N': 44}

Sway with Reuse = False
all  {'Lbs-': 2800.0, 'Acc+': 15.5, 'Mpg+': 20.0, 'N': 398}
     {'Lbs-': 887.21, 'Acc+': 2.71, 'Mpg+': 7.75, 'N': 398}

best {'Lbs-': 2158.0, 'Acc+': 15.5, 'Mpg+': 30.0, 'N': 11}
     {'Lbs-': 307.36, 'Acc+': 2.13, 'Mpg+': 3.88, 'N': 11}

rest {'Lbs-': 2855.0, 'Acc+': 15.3, 'Mpg+': 20.0, 'N': 44}
     {'Lbs-': 846.51, 'Acc+': 2.56, 'Mpg+': 7.75, 'N': 44}

Sway with Halves increased to 1000
all  {'Lbs-': 2800.0, 'Acc+': 15.5, 'Mpg+': 20.0, 'N': 398}
     {'Lbs-': 887.21, 'Acc+': 2.71, 'Mpg+': 7.75, 'N': 398}

best {'Lbs-': 2395.0, 'Acc+': 16.9, 'Mpg+': 30.0, 'N': 13}
     {'Lbs-': 209.3, 'Acc+': 2.13, 'Mpg+': 3.88, 'N': 13}

rest {'Lbs-': 2870.0, 'Acc+': 15.9, 'Mpg+': 20.0, 'N': 52}
     {'Lbs-': 868.22, 'Acc+': 2.4, 'Mpg+': 7.75, 'N': 52}

Sway with min reduced to 0.2
all  {'Lbs-': 2800.0, 'Acc+': 15.5, 'Mpg+': 20.0, 'N': 398}
     {'Lbs-': 887.21, 'Acc+': 2.71, 'Mpg+': 7.75, 'N': 398}

best {'Lbs-': 1995.0, 'Acc+': 15.0, 'Mpg+': 40.0, 'N': 3}
     {'Lbs-': 93.02, 'Acc+': 0.66, 'Mpg+': 0.0, 'N': 3}

rest {'Lbs-': 3190.0, 'Acc+': 14.0, 'Mpg+': 20.0, 'N': 12}
     {'Lbs-': 943.8, 'Acc+': 4.65, 'Mpg+': 7.75, 'N': 12}
```

As seen in the above figure, the values become worse when we disable the reuse factor or when we increase the number of halves to 1000. The values stay almost similar when reducing the min parameter to 0.2 We can, thus, conclude that the reuse and number of halves are very important parameters for this dataset but the min parameter is not as important.

B. ***HPO Study***

An HPO (Hyper-parameter optimization) study involves optimizing the hyperparameters of a machine learning model to improve its performance on a given task. This is done by systematically searching through a predefined hyperparameter space and evaluating the model's performance using a validation set. The goal is to find the best set of hyperparameters that minimize a chosen objective function, such as maximizing accuracy or minimizing loss. HPO studies can be performed using various techniques, such as random search, grid search, and Bayesian optimization, and can significantly improve a model's performance compared to using default hyperparameters.

To carry this out, a typical workflow involves performing initial sampling, then training a model and optimizing it based on metrics like accuracy and recall. Finally, an optimization model are used and we leveraged the ”hyperopt” model for this stage. We were able to realize that minimal sampling methods can be quite effective for learning good Z values.

V. Discussion

A. Threats to Validity

The validity of our approach could be questioned when there is variability in the results which could be attributed to certain datasets with unique distributions and characteristics, including those that have significant levels of noise or invalid data. The performance of the proposed method may be influenced by the properties of the datasets being analyzed. K-means clustering can be sensitive to the initial selection of centroids, leading to different clustering results. This could affect the validity of the results as different initializations may lead to different conclusions. Decision trees can easily overfit the data if the tree is too complex, which could lead to poor generalization performance. This could threaten the validity of the findings as the overfitted model may not be suitable for new, unseen data.

B. Future Work

In future work, better algorithms could be explored for the optimization of SWAY and XLPN. One such algorithm could be DBSCAN, which is a density-based clustering algorithm that is less sensitive to initialization than K-means and is suitable for datasets with irregular shapes and varying densities. Another potential algorithm is Random Forest, which is an ensemble learning technique that can overcome the overfitting issue of Decision Trees and provide better generalization performance. Additionally, other optimization techniques such as genetic algorithms or simulated annealing could be investigated to explore the search space more thoroughly and potentially identify better solutions. Overall, the exploration of better algorithms could lead to more robust and reliable results for SWAY and XLPN optimization.

VI. Conclusion

From our study on the `SWAY` and `XPLN` techniques, we found some interesting results. We were able to realize the cost-effectiveness of different algorithms and were also able to compare the performance of these algorithms across a varying bunch of datasets. Barring a few anomalies, we were able to see the effectivness of our solutions and gather insights on the algorithms' performance. In addition to those, we carried out an ablation study and a HPO study which further enhanced our understanding of the fundamentals of these two techniques.

References

[1] Ghasemi, Zahra, Hadi Akbarzadeh Khorshidi, and Uwe Aickelin. ”Multi-objective Semi-supervised clustering for finding predictive clusters.” Expert Systems with Applications 195 (2022): 116551.

[2] Alok, Abhay Kumar, Sriparna Saha, and Asif Ekbal. ”A new semi-supervised clustering technique using multi-objective optimization.” Applied Intelligence 43 (2015): 633-661.

[3] A. S. Sayyad, T. Menzies, and H. Ammar, “On the value of user preferences in search-based software engineering: A case study in software product lines,” in 2013 35Th international conference on software engineering (ICSE). IEEE, 2013, pp. 492–501.

[4] E. Zitzler and S. Kunzli, “Indicator-based selection in ¨ multiobjective search,” in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 832–842. [Online]. Available: https://doi.org/10.1007/978-3-540-30217-9 84

[5] T. Wagner, N. Beume, and B. Naujoks, “Pareto-, aggregation- , and indicator-based methods in many-objective optimization,” in Evolutionary Multi-Criterion Optimization, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 742–756.

[6] J. Chen, V. Nair, R. Krishna and T. Menzies, ”“Sampling” as a Baseline Optimizer for Search-Based Software Engineering,” in IEEE Transactions on Software Engineering, vol. 45, no. 6, pp. 597-614, 1 June 2019, doi: 10.1109/TSE.2018.2790925.